1. A variable latency cache memory, comprising:

an input, for specifying a type of an instruction requesting to read data from the cache memory, wherein said type is one of a plurality of predetermined instruction types; and

a plurality of storage elements, coupled to said input, for providing said data in a first number of clock cycles if said input specifies a first predetermined one of said plurality of predetermined instruction types, and for providing said data in a second number of clock cycles if said input specifies a second predetermined one of said plurality of predetermined instruction types, wherein said first and second number of clock cycles is different.

2. The cache memory of claim 1, wherein said plurality of storage elements is configured as a last-in-first-out (LIFO) memory.

3. The cache memory of claim 1, further comprising:

a second plurality of storage elements, coupled to said first plurality of storage elements, for caching non-stack data, whereas said first plurality of storage elements is for caching stack data.

4. The cache memory of claim 3, wherein said second plurality of storage elements provides said data in a third number of clock cycles if said input specifies said second predetermined one of said plurality of predetermined instruction types, wherein said second and third number of clock cycles is different.

5. The cache memory of claim 4, wherein said third number of clock cycles is greater than said second number of clock cycles.

6. The cache memory of claim 5, wherein said second predetermined one of said plurality of predetermined instruction types comprises a load instruction type.

7. The cache memory of claim 1, wherein said first predetermined one of said plurality of predetermined instruction types comprises a pop instruction type, wherein said second predetermined one of said plurality of predetermined instruction types comprises a load instruction type.

8. The cache memory of claim 1, wherein a computer data signal embodied in a transmission medium comprising computer-readable program code provides the cache memory.

9. The cache memory of claim 1, a computer program product comprising a computer usable medium having computer readable program code causes the cache memory, wherein said computer program product is for use with a computing device.

10. A variable latency cache memory, comprising:

a plurality of storage elements, configured as a last-in-first-out (LIFO) memory, having first and second subsets of said plurality of storage elements, said first subset for caching stack data more recently pushed than data cached in said second subset;

an input, for specifying an address of source data requested from the cache memory; and

at least one comparator, coupled to said input, for comparing said address with one or more addresses of said data cached in said first subset of storage elements, wherein if said address hits in said first subset based on said comparing, the cache memory provides said source data from said first subset in a first number of clock cycles, wherein if said address does not hit in said first subset based on said comparing, the cache memory provides said source data in a second number of clock cycles, wherein said first and second number of clock cycles is different.

11. The cache memory of claim 10, wherein if said address does not hit in said first subset based on said comparing, the cache memory provides said source data in a second number of clock cycles from said second subset if said address hits in said second subset.

12. The cache memory of claim 10, wherein said address comprises a virtual address.

13. The cache memory of claim 12, further comprising:

a second input, coupled to said plurality of storage elements, for specifying a physical address of said source data requested from the cache memory.

14. The cache memory of claim 13, further comprising:

a second at least one comparator, coupled to receive said second input, for comparing said physical address with one or more physical addresses of said data cached in said first subset of storage elements.

15. The cache memory of claim 14, further comprising:

an output, coupled to said first and second at least one comparator, for indicating an error condition if the cache memory provides said source data from said first subset based on said comparing said virtual address with one or more addresses of said data cached in said first subset of storage elements, but said second at least one comparator indicates said physical address does not match any of said one or more physical addresses of said data cached in said first subset of storage elements.

16. The cache memory of claim 10, further comprising:

a second plurality of storage elements, coupled to said first plurality of storage elements, for caching non-stack data.

17. The cache memory of claim 16, wherein said second plurality of storage elements provides said data in a third number of clock cycles if said address does not hit in said first plurality of storage elements, wherein said second and third number of clock cycles is different.

18. The cache memory of claim 17, wherein said third number of clock cycles is greater than said second number of clock cycles.

19. The cache memory of claim 10, wherein if said address does not hit in said first subset based on said comparing, the cache memory provides said source data in a second number of clock cycles based on a physical address compare.

20. The cache memory of claim 10, wherein said first subset comprises a top one of said plurality of storage elements.

21. The cache memory of claim 10, wherein said first subset comprises a top two of said plurality of storage elements.

22. The cache memory of claim 10, wherein said first subset comprises a top three of said plurality of storage elements.

23. The cache memory of claim 10, wherein said address comprises a source address of a load instruction.

24. The cache memory of claim 10, wherein a computer data signal embodied in a transmission medium comprising computer-readable program code provides the cache memory.

25. The cache memory of claim 10, a computer program product comprising a computer usable medium having computer readable program code causes the cache memory, wherein said computer program product is for use with a computing device.

26. A method for providing data from a cache memory with a variable latency, the method comprising:

storing stack data into the cache memory in a last-in-first-out manner;

providing load data from the cache memory in a first number of clock cycles if a virtual address of the load data hits in the cache memory; and

providing the load data from the cache memory in a second number of clock cycles if the virtual address of the load data misses in the cache memory but a physical address of the load data hits in the cache memory, wherein the first and second number of clock cycles is different.

27. The method of claim 26, further comprising:

determining whether the virtual address hits in a top subset of cache lines of the cache memory, wherein the top subset is less than all cache lines of the cache memory;

wherein said providing the load data from the cache memory in a first number of clock cycles if a virtual address of the load data hits in the cache memory is in response to said determining.

28. The method of claim 27, wherein the top subset of cache lines of the cache memory comprises cache lines implicated by most recently pushed stack data.

29. The method of claim 26, wherein the first number of clock cycles is less than the second number of clock cycles.

30. The method of claim 26, wherein said providing the load data from the cache memory in the first number of clock cycles if the virtual address of the load data hits in the cache memory is speculative subject to a subsequent determination that the physical address of the load data hits in the cache memory.

31. The method of claim 26, further comprising:

providing the load data from a non-stack cache memory in a third number of clock cycles if the virtual address and the physical address miss in the cache memory, wherein the first and third number of clock cycles is different.

32. A method for providing data from a cache memory with a variable latency, the method comprising:

determining whether a request for data from the cache memory is in response to a pop or load instruction;

providing the data in a first number of clock cycles if the request is in response to a pop instruction; and

providing the data in a second number of clock cycles if the request is in response to a load instruction, wherein the first and second number of clock cycles is different.

33. The method of claim 32, wherein the first number of clock cycles is less than the second number of clock cycles.

34. The method of claim 32, wherein said providing the data in the first number of clock cycles if the request is in response to a pop instruction is speculative subject to a subsequent determination that a source address of the data hits in the cache memory.

35. The method of claim 32, wherein a load instruction comprises an instruction explicitly specifying a source address of the data.

36. The method of claim 32, wherein a pop instruction comprises an instruction inherently specifying a source address of the data.

37. The method of claim 36, wherein the pop instruction inherently specifies the source address of the data relative to a stack pointer value.

38. A computer data signal embodied in a transmission medium, comprising:

computer-readable program code for providing a variable latency cache memory, said program code comprising:

first program code for providing an input, for specifying a type of an instruction requesting to read data from the cache memory, wherein said type is one of a plurality of predetermined instruction types; and

second program code for providing a plurality of storage elements, coupled to said input, for providing said data in a first number of clock cycles if said input specifies a first predetermined one of said plurality of predetermined instruction types, and for providing said data in a second number of clock cycles if said input specifies a second predetermined one of said plurality of predetermined instruction types, wherein

said first and second number of clock cycles is different.

39. A computer data signal embodied in a transmission medium, comprising:

computer-readable program code for providing a variable latency cache memory, said program code comprising:

first program code for providing a plurality of storage elements, configured as a last-in-first-out (LIFO) memory, having first and second subsets of said plurality of storage elements, said first subset for caching stack data more recently pushed than data cached in said second subset;

second program code for providing an input, for specifying an address of source data requested from the cache memory; and

third program code for providing at least one comparator, coupled to said input, for comparing said address with one or more addresses of said data cached in said first

subset of storage elements, wherein if said address hits in said first subset based on said comparing, the cache memory provides said source data from said first subset in a first number of clock cycles, wherein if said address does not hit in said first subset based on said comparing, the cache memory provides said source data in a second number of clock cycles, wherein said first and second number of clock cycles is different.